a data retriever mechanism in communication with the data handler mechanism, the data retriever mechanism being arranged to obtain the data and to pass the data to the data handler mechanism; and

a mapping mechanism in communication with the data handler mechanism, the mapping mechanism being substantially separate from the data handler mechanism, the mapping mechanism being arranged to obtain the command object and provide a command list identifying commands associated with the data to the plurality of applications for display, the command list provided by using a data handler mechanism, wherein the mapping mechanism is associated with the plurality of applications and is arranged to obtain the command object without directly involving the selected application.

26. A computer-implemented framework as recited in claim [1]23 wherein the mapping mechanism and the data handler mechanism are separately maintained.

27. A computer-implemented framework as recited in claim [1]23 wherein the mapping mechanism is not specific to the application while the data handler mechanism is substantially specific to the application.

REMARKS

Claims 1-27 were rejected under 35 U.S.C. §103(a) as being unpatentable over European Patent Publication EP 0 718 761 A1 (Gosling). The Applicants acknowledge the Examiner's assertion that Gosling III does not explicitly mention that the data handler mechanism allows use of new command objects without modifying the application. Claims 1, 9, 16, and 23 have been amended to clarify the invention. Claims 21, 22, 26, and 27 have been amended to make typographical changes.

Eight office actions associated with the present case have been issued. A CPA as well as an RCE have been filed. The Applicants believe the claims are allowable in their present form, as Gosling neither sufficient teaches or suggests that a "data handler mechanism" allows use of new command objects without modifying the application. However, because of the large number of actions issued in this case, amendments to the independent claims have been made to facilitate prosecution.

Independent claims 1, 9, 16, and 23 now variably recite a mapping mechanism being arranged to obtain a command list identifying commands associated with the data and providing the command list through the data handler mechanisms to an application for display. Support for the claims can be found on page 16 lines 12-29 as well as in Figure 6. More specifically, "the data handler queries the command map for the command list that is associated with the returned type. In general, it should be appreciated that each data type has an associated command list... The command map returns the command list associated with the data type to the data handler... After the data handler has obtained the command list, the data handler returns the command list to the application... and the application displays the command list."

Gosling is believed not to not teach or suggest a command map obtaining a command list. Gosling does not disclose providing the command list to the application by using a data handler. Furthermore, Gosling makes no mention of displaying the command list. In fact, Gosling does not even mention a command list or commands associated with data.

In view of the above, Applicants believe that all pending claims are allowable and respectfully request a Notice of Allowance for this application from the Examiner. If the Examiner has any questions or concerns, please feel free to contact the undersigned at the telephone number set out below.

If any fees are due in connection with the filing of this amendment, the Commissioner is authorized to charge such fees to Deposit Account 50-0388 (Order No. SUN1P123). A duplicate copy of the transmittal is enclosed for this purpose.

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP

Godfrey K. Kwan
Reg. No. 46,850

P.O. Box 778
Berkeley, California 94704-0778
(510) 843-6200

APPENDIX A – ALL PENDING CLAIMS (MARKED-UP VERSION)

1. (Amended) A computer-implemented framework for associating data with a command object, the command object being arranged to operate on the data, wherein the data is associated with an application, the computer-implemented framework comprising:

a data handler mechanism arranged as a generic interface with the application, wherein the generic interface allows use of new command objects without modifying the application;

a data retriever mechanism in communication with the data handler mechanism, the data retriever mechanism being arranged to obtain the data and to pass the data to the data handler mechanism; and

a mapping mechanism in communication with the data handler mechanism, the mapping mechanism being separate from the data handler mechanism, the mapping mechanism being arranged to obtain the command object and a command list identifying commands associated with the data, wherein the command object is obtained by the mapping mechanism based substantially on the data and wherein the command list is returned through the data handler mechanism to the application for display.

2. A computer-implemented framework according to claim 1 wherein the data is a stream of bytes, and the data handler mechanism is further arranged to bind the stream of bytes to the command object.

3. A computer-implemented framework according to claim 1 wherein the data retriever mechanism includes a data content handler mechanism in communication with the data handler mechanism, the data content handler mechanism being arranged to convert the data into a data object, wherein the data handler mechanism is further arranged to bind the data object to the command object.

4. A computer-implemented framework as recited in claim 3 wherein the data object is created using the Java™ programming language, and the command object is a Java™ command object.

5. A computer-implemented framework as recited in claim 1 wherein the data is one of text data and image data.

6. A computer-implemented framework as recited in claim 1 wherein the data handler is further arranged to receive a request from the application, to bind the data to the command object, and to return the command object to the application.

7. A computer-implemented framework as recited in claim 1 wherein the data retriever includes a data source mechanism arranged to obtain a stream of bytes and a data content handler mechanism arranged to convert the stream of bytes into a data object, the data source mechanism being in communication with the data content handler mechanism, wherein the data handler mechanism is further arranged to bind the data object to the command object.

8. A computer-implemented framework as recited in claim 1 wherein the mapping mechanism includes a look-up table arranged to associate the command object with the data.

9. (Amended) A computer-implemented method for associating data with a command object in response to a request from an application, the method comprising:

accessing the data through an interface in response to the request from the application, the interface being independent from the application and in communication with the application, wherein the request from the application is processed by the interface, the interface allowing use of new command objects without modifying the application;

accessing a mapping mechanism which is in communication with the interface, the mapping mechanism being independent from the application such that the mapping mechanism is not a component of the application, the mapping mechanism being maintained separately from the interface, the mapping mechanism further being arranged to locate a command object that is appropriate for the data, wherein the mapping mechanism is accessed by the interface;

obtaining the command object that is appropriate for the data and a command list identifying commands associated with the data, wherein the mapping mechanism obtains the command object and passes the obtained command object to the interface;

binding the command object to the data, wherein the interface binds the command object to the data; and

returning the command object and the command list to the application, wherein the interface returns the command object to the application and the data handler mechanisms returns the command list to the application for display.

10. A computer-implemented method as recited in claim 9 wherein accessing the data through an interface includes:

passing a stream of bytes to a data content handler mechanism arranged to create a data object from the stream of bytes; and

passing the data object to the interface, wherein the data is the data object.

11. A computer-implemented method as recited in claim 10 wherein the data object is created using the Java™ programming language, and the command object is a Java™ command object.

12. A computer-implemented method as recited in claim 9 wherein accessing the data through the interface includes accessing a data retriever which is arranged to obtain the data, wherein the data is a stream of bytes.

13. A computer-implemented method as recited in claim 9 further including operating on the data using the command object.

14. A computer-implemented method as recited in claim 9 wherein the command object that is appropriate for the data is selected from a set of command objects associated with a command list, the command list being associated with the data, the method further including accessing the command list, wherein the command list is accessed by the interface.

15. A computer-implemented method as recited in claim 14 wherein accessing the command list includes receiving a request for a command list from the application, the request for the command list being received by the interface, wherein the interface performs the steps of:

obtaining a type associated with the data;

obtaining the command list through the mapping; and

returning the command list to the application.

16. (Amended) A computer program product for associating data with a command object in response to a request from an application, the computer program product comprising:

computer code for accessing the data through an interface in response to the request from the application, the interface being independent from the application and in communication with the application, wherein the request from the application is processed by the interface, the interface allowing use of new command objects without modifying the application;

computer code for accessing a mapping mechanism which is in communication with the interface, the mapping mechanism being independent from the application such that the mapping mechanism is not a part of the application, the mapping mechanism further being separately maintained from the interface, the mapping mechanism further being arranged to locate a command object that is appropriate for the data, wherein the mapping mechanism is accessed by the interface;

computer code for obtaining the command object that is appropriate for the data and a command list identifying commands associated with the data, wherein the mapping mechanism obtains the command object and passes the obtained command object to the interface;

computer code for binding the command object to the data, wherein the interface binds the command object to the data; and

computer code for returning the command object and the command list to the application, wherein the interface returns the command object to the application and the data handler mechanisms returns the command list to the application for display.

a computer-readable medium that stores the computer codes.

17. A computer-readable medium as recited in claim 16 wherein the computer program code devices configured to cause the computer to access the data through an interface include computer program code devices configured to cause a computer to execute the steps of:

passing a stream of bytes to a data content handler mechanism arranged to create a data object from the stream of bytes; and

passing the data object to the interface, wherein the data is the data object.

18. A computer-readable medium as recited in claim 17 wherein the data object is created using the Java™ programming language, and the command object is a Java™ command object.

19. A computer-readable medium as recited in claim 16 further including computer program code devices configured to cause the computer to operate on the data using the command object.

20. A computer-readable medium as recited in claim 16 wherein the command object that is appropriate for the data is selected from a set of command objects associated with a command list, the command list being associated with the data, the computer-readable medium further including computer code devices configured to cause the computer to access the command list through the interface.

21. A computer-implemented framework according to claim 16 wherein the command object is obtained by the mapping mechanism based substantially on the data without an external input from a user of the application.

22. A computer-implemented framework according to claim 16 wherein the command object is obtained by the mapping mechanism based substantially on the data without directly involving the application.

23. (Amended) A computer-implemented framework for associating data with a command object, the command object being arranged to operate on the data, wherein the data is associated with a selected application, the computer-implemented framework comprising:

a data handler mechanism arranged to interface with a plurality of applications, the plurality of applications including the selected application, wherein the data handler mechanism is independent from the plurality of applications and allows use of new command objects without modifying the application;

a data retriever mechanism in communication with the data handler mechanism, the data retriever mechanism being arranged to obtain the data and to pass the data to the data handler mechanism; and

a mapping mechanism in communication with the data handler mechanism, the mapping mechanism being substantially separate from the data handler mechanism, the mapping mechanism being arranged to obtain the command object and provide a command list identifying commands associated with the data to the plurality of applications for display, the command list provided by using a data handler mechanism, wherein the mapping mechanism is associated with the plurality of applications and is arranged to obtain the command object without directly involving the selected application.

24. A computer-implemented framework as recited in claim 23 wherein the mapping mechanism and the data handler mechanism are separately maintained.

25. A computer-implemented framework as recited in claim 23 wherein the mapping mechanism is not a component of the data handler mechanism.

26. A computer-implemented framework as recited in claim [1]23 wherein the mapping mechanism and the data handler mechanism are separately maintained.

27. A computer-implemented framework as recited in claim [1]23 wherein the mapping mechanism is not specific to the application while the data handler mechanism is substantially specific to the application.